Equation: Momentum & Adaptive LR

Ki Hyun Kim

nlp.with.deep.learning@gmail.com

Review: SGD

- Learning rate is a hyper-parameter that need to be tuned.

$$\mathcal{L}(\theta_t) = \frac{1}{N}\sum_{i=1}^{N}\Delta\Big(f(x_i;\theta_t), y_i\Big)$$

$$g_t = \nabla_\theta \mathcal{L}(\theta_t)$$

$$\theta_{t+1} = \theta_t - \eta \cdot g_t,$$

where η is learning rate.

Momentum

- Accumulate gradient from the beginning with discount factor.

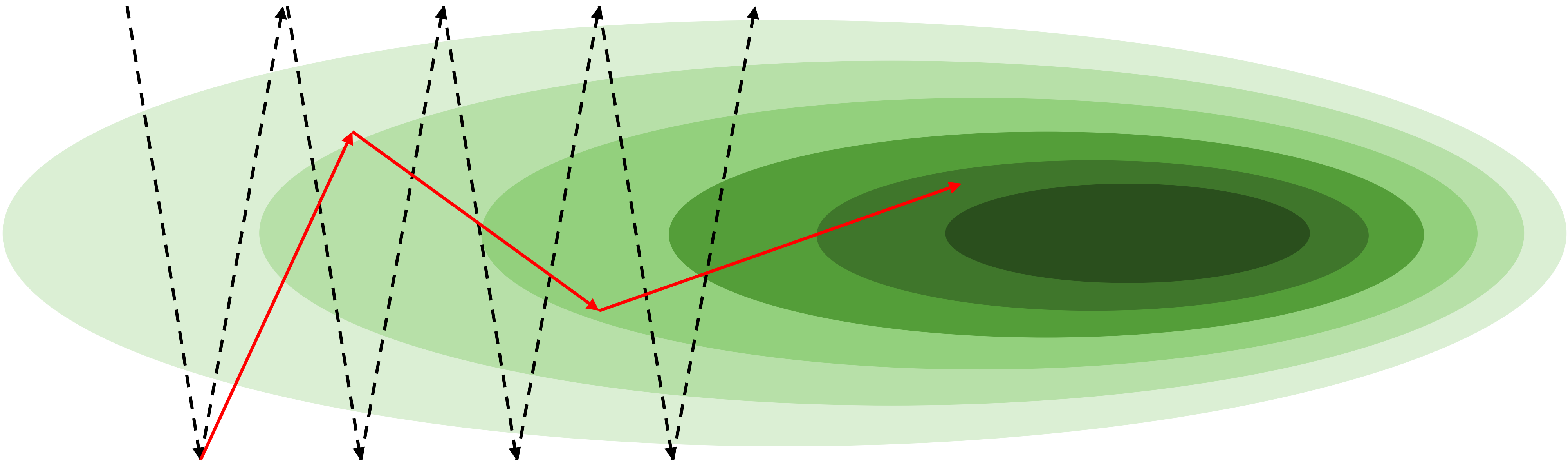
$$\begin{aligned}\tilde{g}_t &= \gamma \cdot \tilde{g}_{t-1} - \eta \cdot g_t \\ &= -\eta \cdot \sum_{i=1}^{t} \gamma^{t-i} \cdot g_i,\end{aligned}$$

where $\tilde{g}_0 = 0$ and γ is momentum.

$$\begin{aligned}\theta_{t+1} &= \theta_t + \tilde{g}_t \\ &= \theta_t - \eta \cdot \sum_{i=1}^{t} \gamma^{t-i} \cdot g_i\end{aligned}$$

Momentum Example

- 깊은 계곡



Adaptive LR Motivation

- 학습 초반에는 큰 LR, 후반에는 작은 LR으로 최적화

- Motivation
  - 학습 초반의 너무 작은 learning rate는 진행이 더디게 되고,
  - 학습 후반의 너무 큰 learning rate는 더 좋은 loss를 얻지 못하게 됨

- 방법
  1) 현재 epoch에서 loss가 과거 epoch의 loss보다 더 나아지지 않을 경우, 일정 비율(보통 0.5)으로 decay.
  2) 정해진 epoch가 지날 때마다 일정 비율로 decay

Adaptive LR: AdaGrad

- Each parameter has its own learning rate, which is divided by sum of squares.

$$
\begin{aligned}
r_t &= r_{t-1} + g_t \odot g_t \\
&= \sum_{i=1}^{t} g_i \odot g_i,
\end{aligned}
$$

where $r_0 = 0$ and $\odot$ is element-wise multiplication.

$$
\begin{aligned}
\theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{r_t + \epsilon}} \odot g_t \\
&= \theta_t - \eta \cdot \frac{g_t}{\sqrt{\epsilon + \sum_{i=1}^{t} g_i \odot g_i}}
\end{aligned}
$$

Adaptive LR + Momentum: Adam

- Two Hyper-params $\rho_1 = 0.9 \text{ and } \rho_2 = 0.999.$

$$
\begin{aligned}
s_t &= \rho_1 \cdot s_{t-1} + (1-\rho_1) \cdot g_t, \text{ where } s_0 = 0. \\
&= (1-\rho_1) \cdot \sum_{i=1}^{t} \rho_1^{t-i} \cdot g_i
\end{aligned}
$$

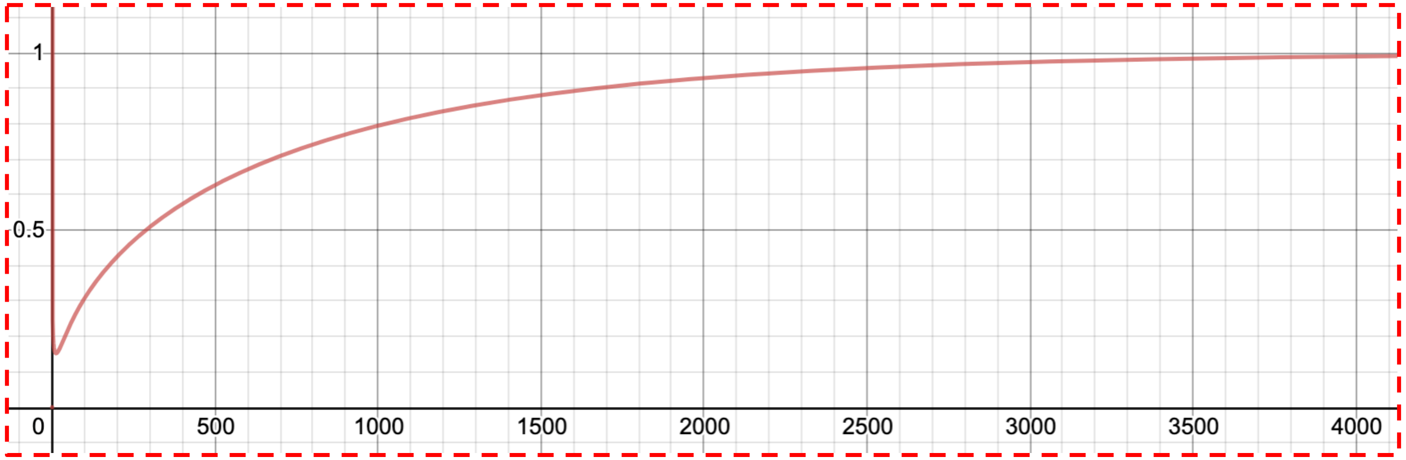
$$
\begin{aligned}
r_t &= \rho_2 \cdot r_{t-1} + (1-\rho_2) \cdot (g_t \odot g_t), \text{ where } r_0 = 0. \\
&= (1-\rho_2) \cdot \sum_{i=1}^{t} \rho_2^{t-i} \cdot (g_i \odot g_i)
\end{aligned}
$$

$$
\hat{s}_t = \frac{s_t}{1-\rho_1^t}
$$

$$
\hat{r}_t = \frac{r_t}{1-\rho_2^t}
$$

$$
\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{s}_t}{\sqrt{\hat{r}_t + \epsilon}}
$$

Adam Explanation



$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{s}_t}{\sqrt{\hat{r}_t + \epsilon}}$$

$$\approx \theta_t - \eta \cdot \frac{\hat{s}_t}{\sqrt{\hat{r}_t}}$$

$$= \theta_t - \eta \cdot \boxed{\frac{\sqrt{1-\rho_2^t}}{1-\rho_1^t}} \cdot \frac{1-\rho_1}{\sqrt{1-\rho_2}} \cdot \frac{\overbrace{\sum_{i=1}^{t} \rho_1^{t-i} \cdot g_i}^{\text{Momentum}}}{\underbrace{\sqrt{\sum_{i=1}^{t} \rho_2^{t-i} \cdot (g_i \odot g_i)}}_{\text{Adaptive LR}}}$$

Wrap-up

- Adam이 가장 hyper-parameter의 변화에 강인(robust)하다고 알려져 있으나, 상황에 따라 가장 알맞은 optimizer를 찾아 활용하는 것이 중요